Recent Progress in Communication-Avoiding Krylov Subspace Methods

Erin Carson and Nick Knight

Motivation

- Krylov Subspace Methods are commonly used for solving linear systems.
- Standard implementations are communication-bound due to required SpMV and orthogonalization in every iteration.
- Solution: rearrange algorithms to perform s iterations at a time without communicating (s-step methods).
- SpMV in each iteration is replaced with a call to the Matrix Powers Kernel, which performs s SpMVs while reading the matrix only once.
- Used to generate s basis vectors for the Krylov Subspace.

Convergence Results

- Summary of Results for All Test Matrices:
  - CA variants (generally) maintain stability for s in between 2 and 10 (monomial basis).
  - Reduces communication costs by a factor of s - if s = 10, possible speedup is 10x!
  - In general, as s increases, the number of iterations needed to converge increases.
  - Could be remedied by preconditioning, restarting, extended precision.
  - Must choose s for both performance and stability.
  - In some cases, the CA variant converges smoothly whereas the standard implementation does not.
  - Skipping bad iterates?
  - Preliminary results for Newton Basis on smaller test matrices show similar convergence patterns.

Previous Work

- Communication-Avoiding Kernels
  - Matrix Powers Kernel (one matrix, one input vector).
  - Tall-Skinny QR.
- One-sided Krylov Subspace Methods
  - Conjugate Gradient [Hoemmen, 2010]
  - GMRES [MHDY09]
  - Lanczos [Hoemmen, 2010]

Recent Work in CA-KSMs

- 3 New CA Methods
  - BiCG, Conjugate Gradient Squared (CGS)
  - Main result: CA-BiCGStab:

<table>
<thead>
<tr>
<th>Computation and Communication Costs (for s steps, Sequential Algorithm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original BiCGStab</td>
</tr>
<tr>
<td>Communication</td>
</tr>
<tr>
<td>Computation</td>
</tr>
</tbody>
</table>

- Convergence Results
  - Sequential C implementation – allows for results on larger test matrices.
  - New approach to avoiding communication in the sequential case for stencil-like matrices.
  - Streaming matrix powers.

Stencil-Like Matrices: Saving Bandwidth in the Sequential Algorithm

- What if A is stencil-like (in general, O(n) cost to read)?
  - In the sequential algorithm:
    - Not communication-bound due to reading A, but…
    - Communication bottleneck is now reading Krylov vectors.
    - O(kn) cost to read Krylov basis vectors every k steps.
  - Can we reduce the communication cost of k steps from O(kn) to O(n)?

- Idea: Don’t explicitly store basis vectors.
  - Streaming Matrix Powers: Interleave matrix powers computation and construction of the Gram Matrix G
  - Part i computes G+=V_i V_i^T, discards V_i.
  - Tradeoff: requires two matrix powers invocations, but bandwidth reduced by a factor of k.
  - OK if reading and applying A is inexpensive (e.g., stencil, AMR base case, others?)

- Overall communication reduced from O(kn) to O(n)!

Future Work

- CA Krylov Subspace methods. Next target: BiCGStab(l)
- Evaluate current preconditioning methods and extend CA approach to other classes of preconditioners
- Parallel Implementations / Performance tests
- Additions to Matrix Powers kernel
  - Multiple right hand sides
  - Partitioning that targets matrix powers (e.g., hypergraph)
- Improving stability
  - Other bases for Matrix Powers (Chebyshev), restarted methods, extended precision
- Extending stability theory to CA-KSMs
- Auto-tuning work
  - Incorporation of Matrix Powers into pOSKI (Jong-Ho Byun, et al., UCB)
  - Code generation for Matrix Powers (collaborating with Ras Bodik, Michelle Strout)
- Exploring co-tuning for CA-KSMS (i.e., Matrix Powers and TSQR)